AI & ML

SESSION 13 (SVM)

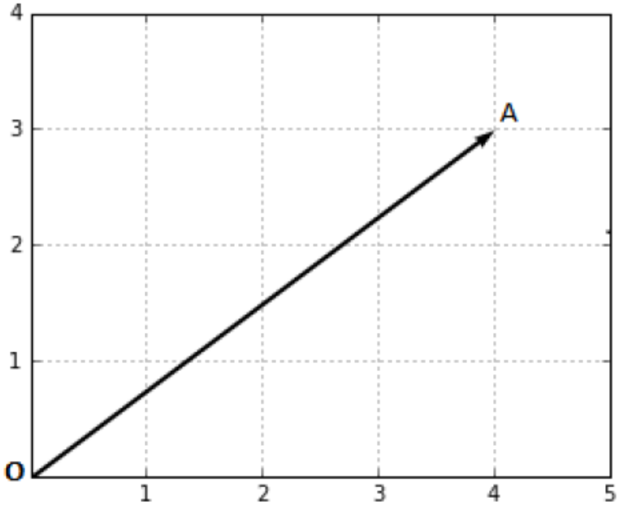
MODULE 3

Session Outline

- Prerequisites
- SVM
- Hard & Soft Margin SVM
- Non-linear SVM
- Multiclass SVM
- Python Implementation

Prerequisites

Vector



$\overrightarrow{OA}$ = (4,3)

a = (4,3)

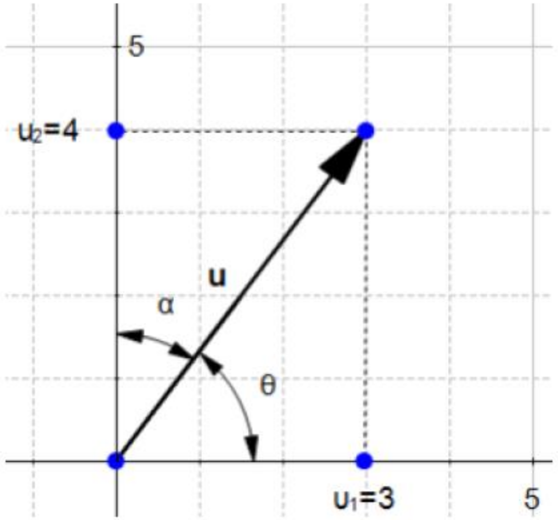
Dimension of a vector ?

Magnitude of vector

$$\|x\| = \sqrt{x_1^2 + x_2^2}$$

Direction of vector

$$u = \left(\frac{x_1}{\|x\|}, \frac{x_2}{\|x\|}\right)$$

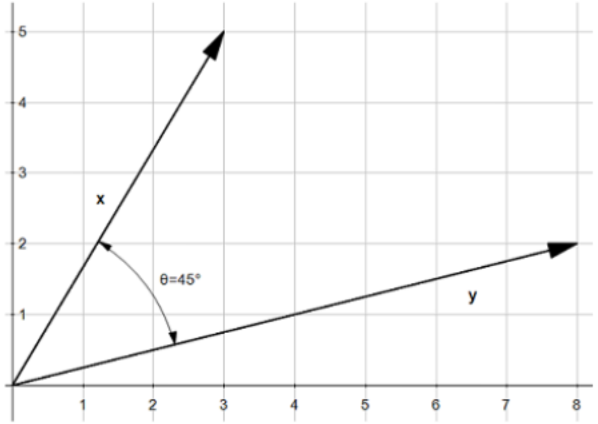


Unit vector = (cos(θ),sin(θ))

The norm of direction vector is always 1

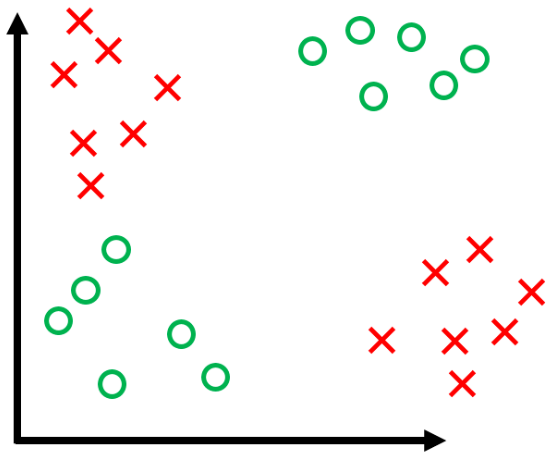
Dot Product

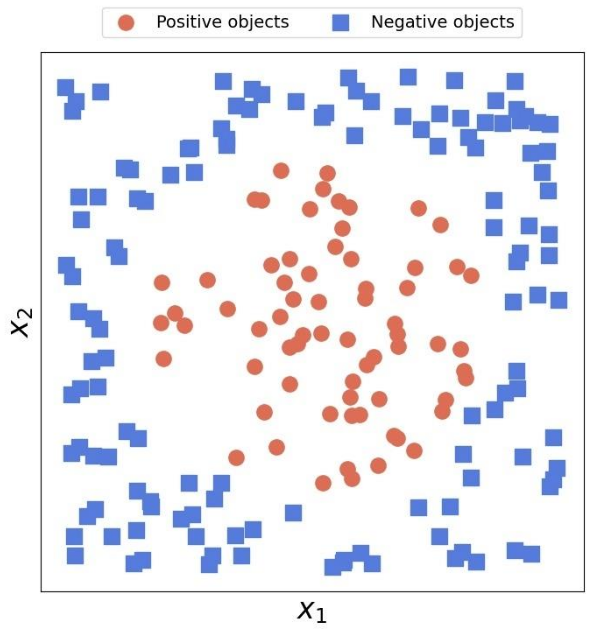
Scalar product

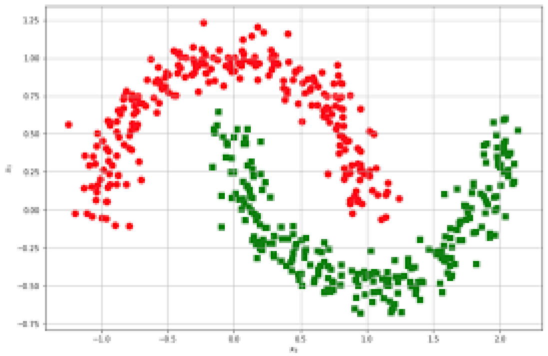


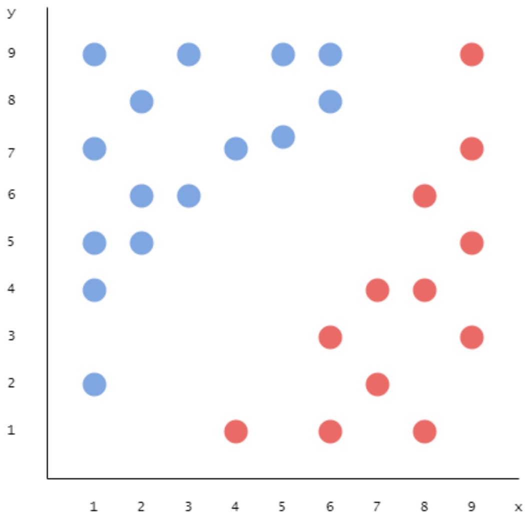
$$\mathbf{x} \cdot \mathbf{y} = ||x||\ ||y||\ cos(\theta)$$

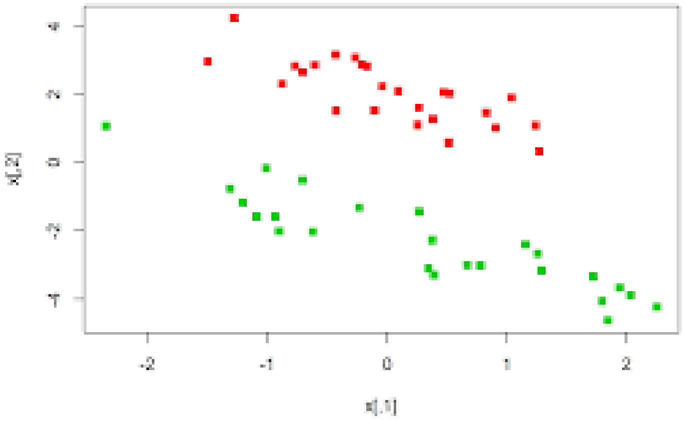
$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n}(x_i y_i)$$

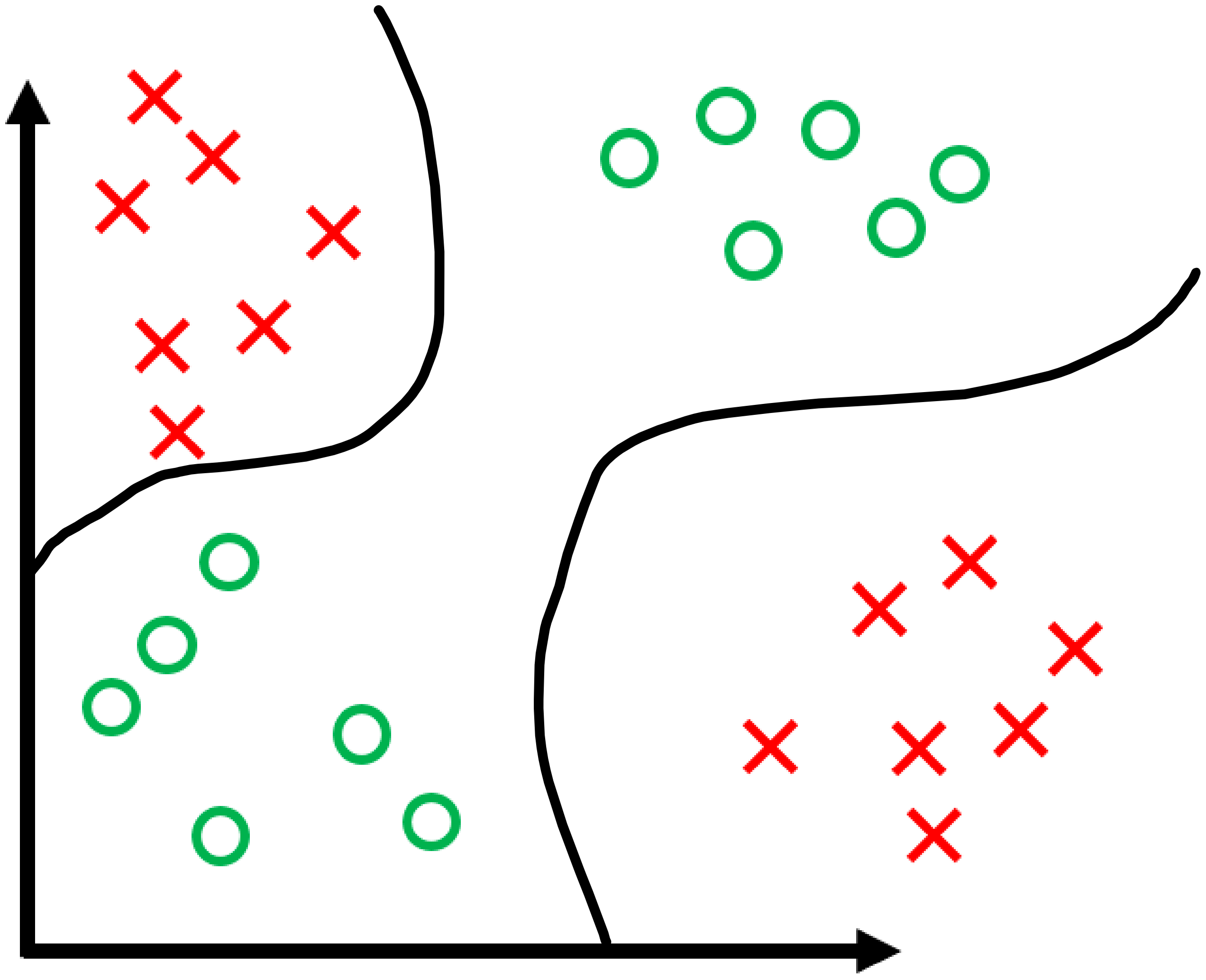


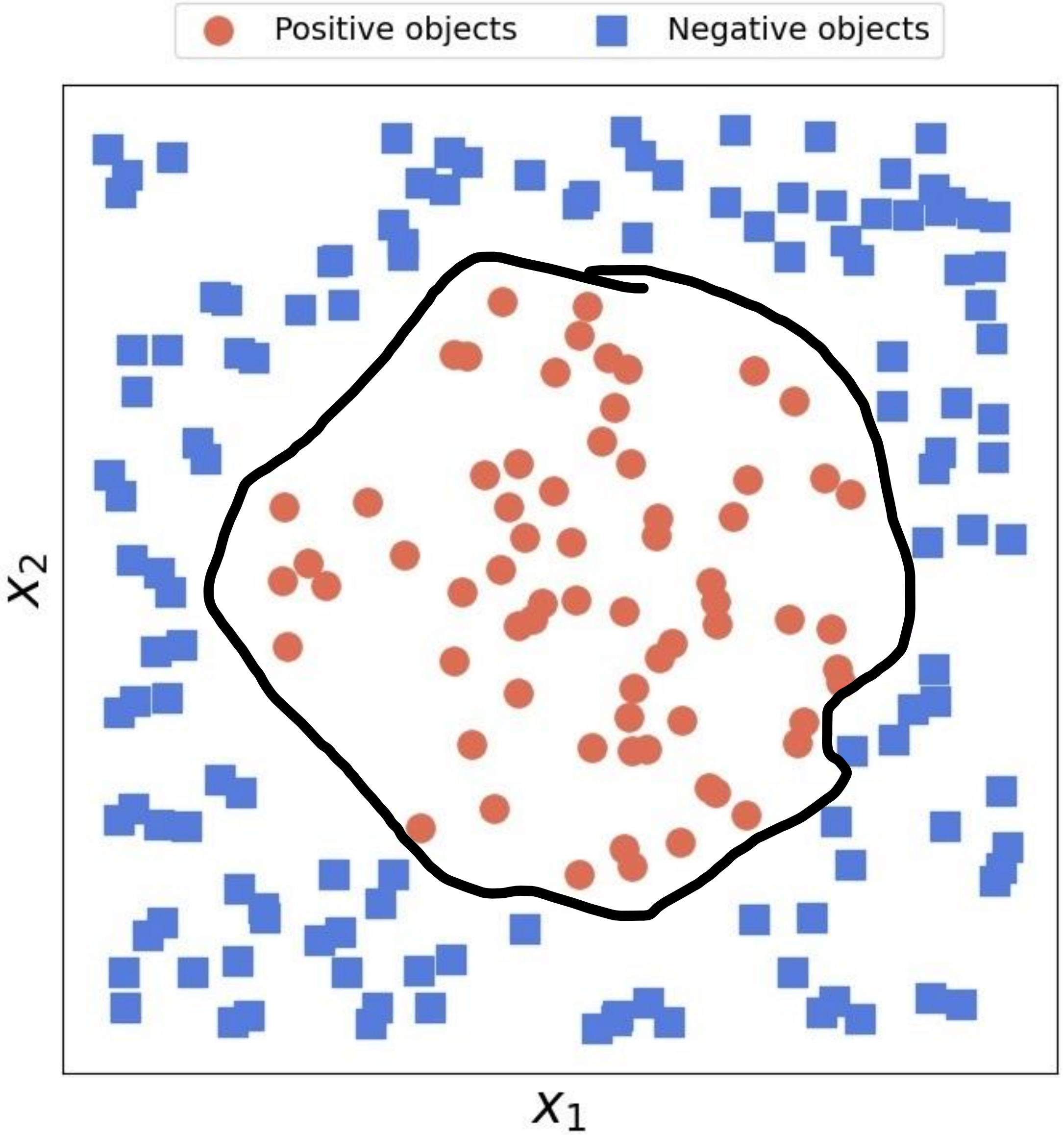

Positive objects
Negative objects
x_2
x_1

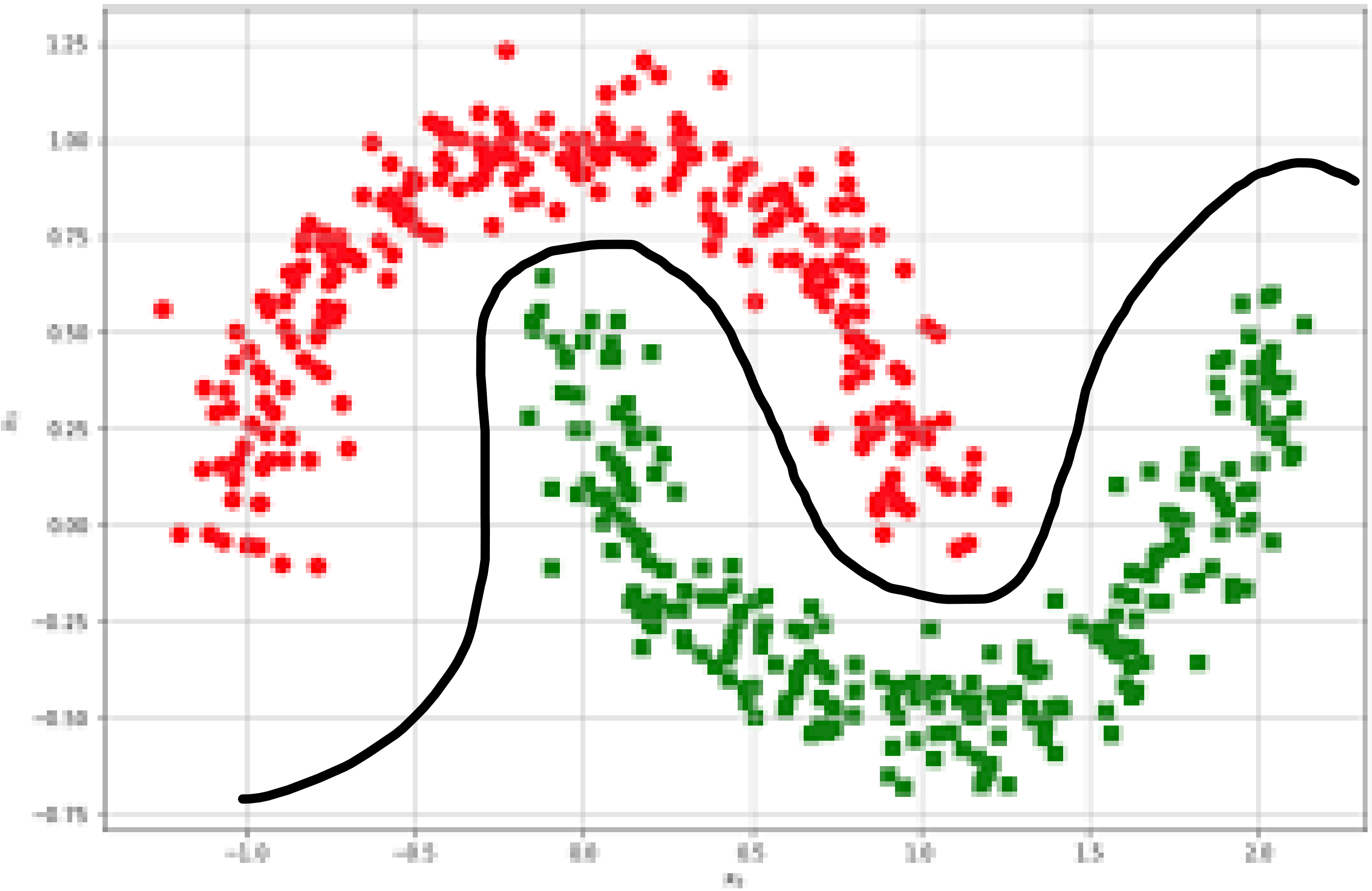


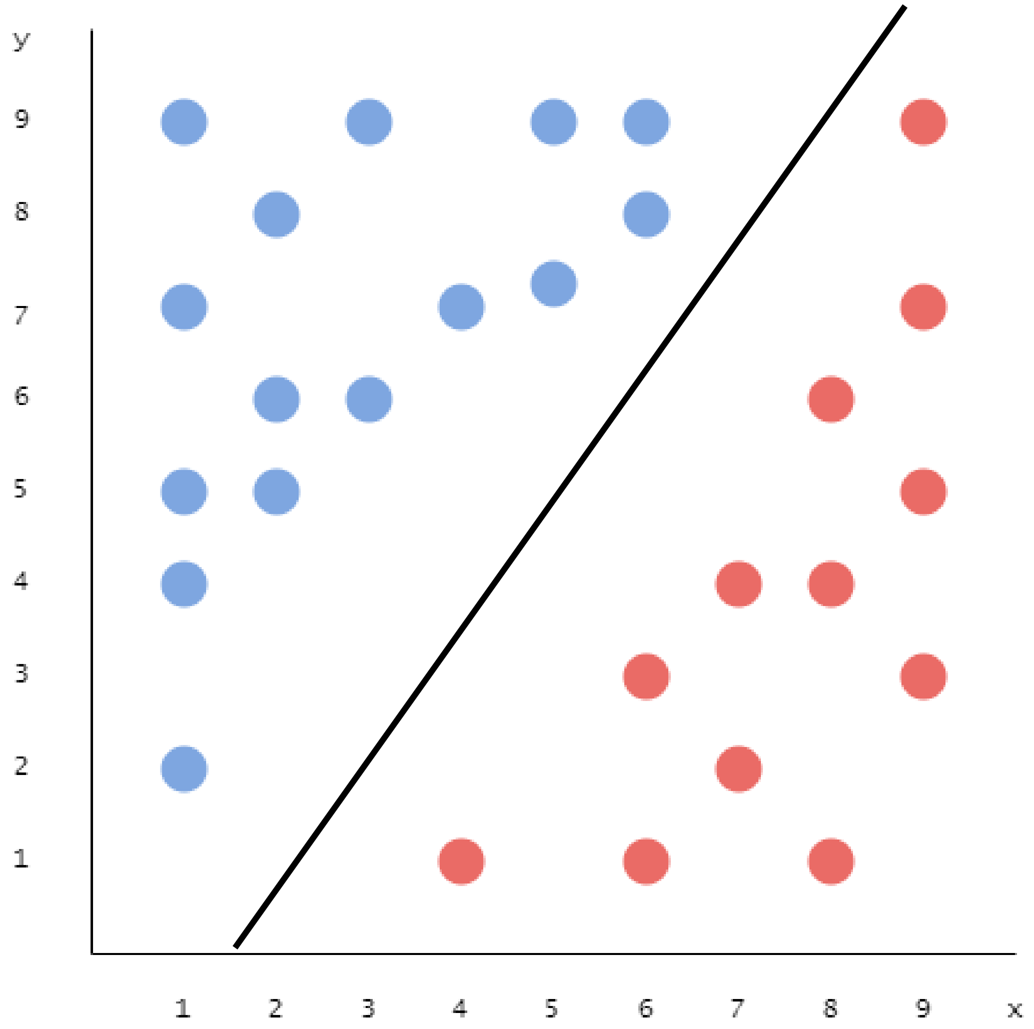

y
9
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8
9
x

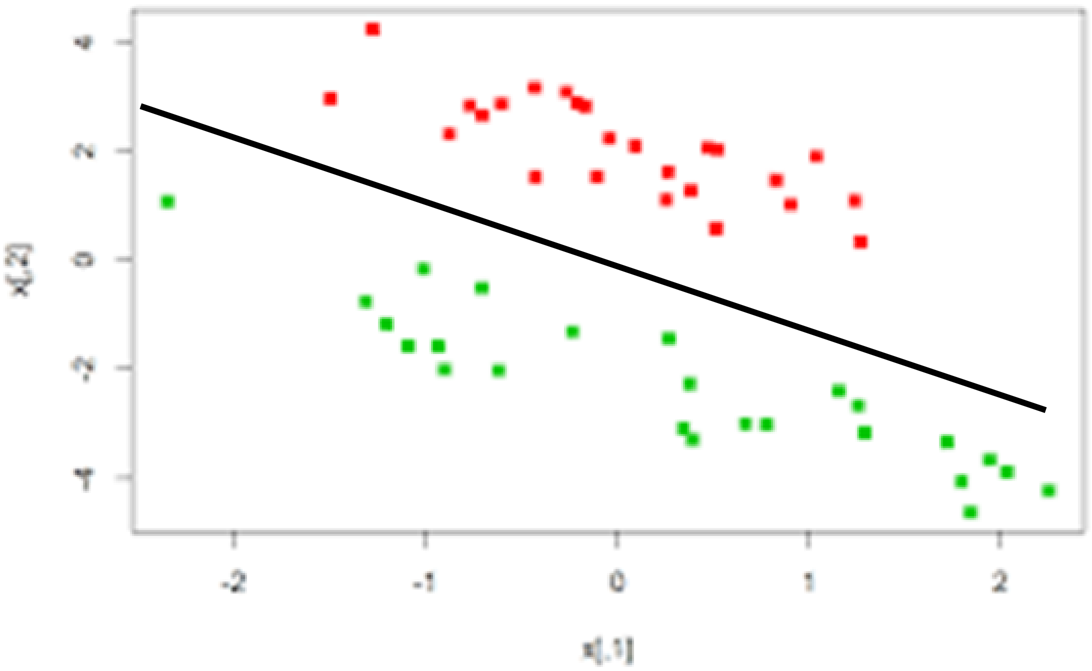

-2
-1
0
1
2



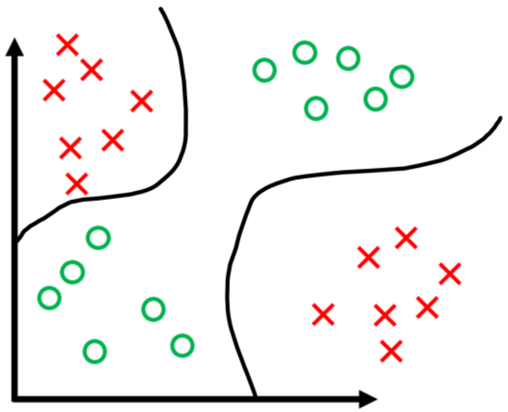

Positive objects
Negative objects
x_2
x_1

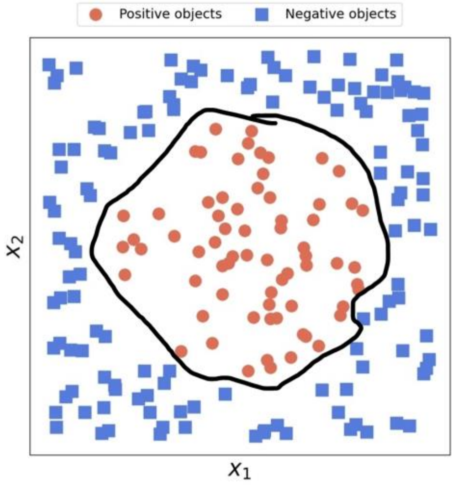

1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

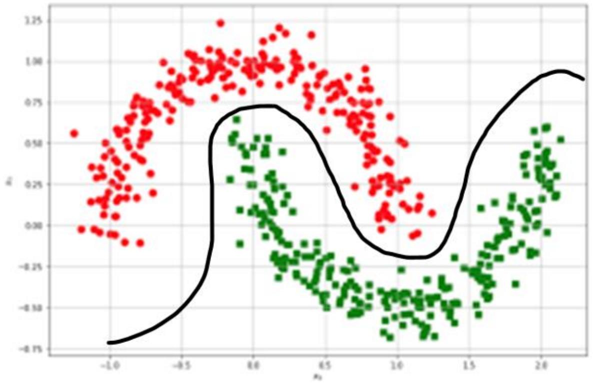

y
9
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8
9
x

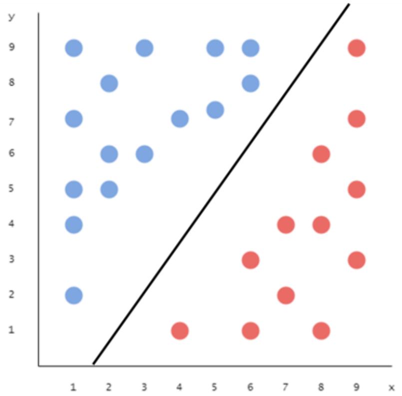


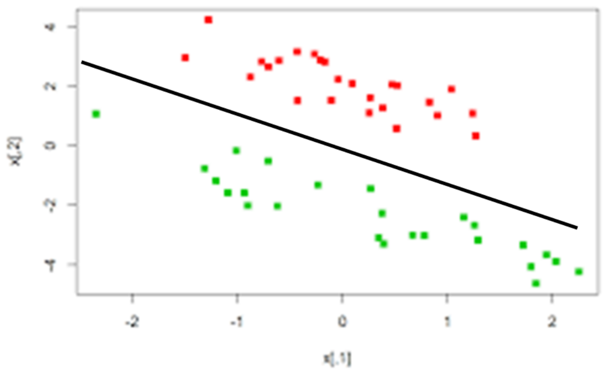
x[,2]
x[,1]




Positive objects
Negative objects
x_2
x_1

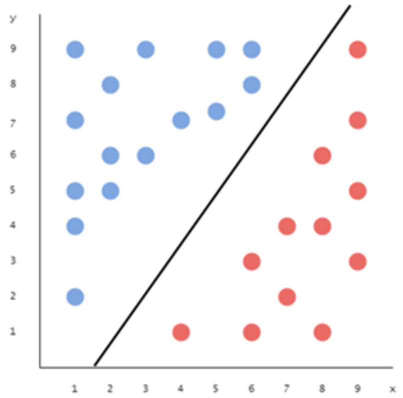


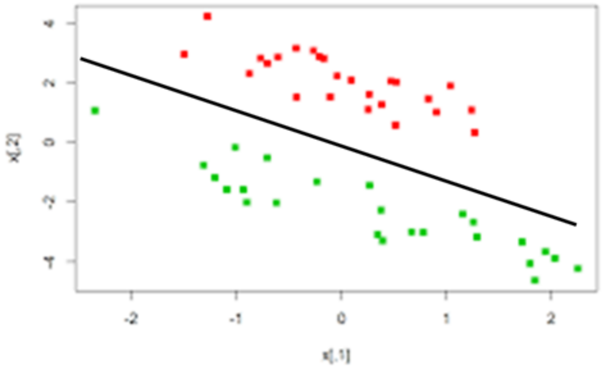

y
9
8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 x


4
2
0
-2
-4
x[,2]
-2 -1 0 1 2
x[,1]

Linearly Separable

A dataset is said to be linearly separable if it is possible to draw a line that can separate the points from each other.

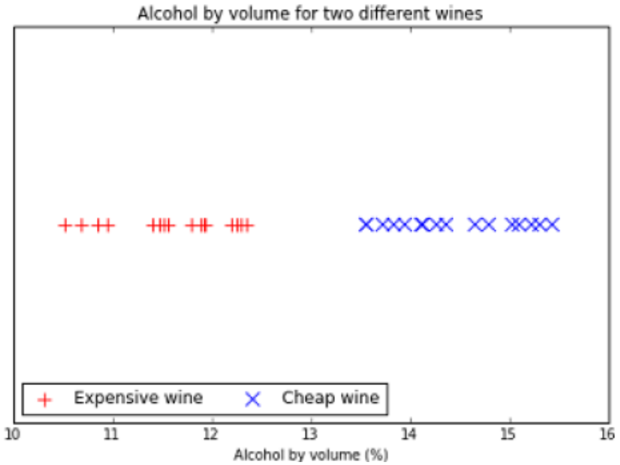




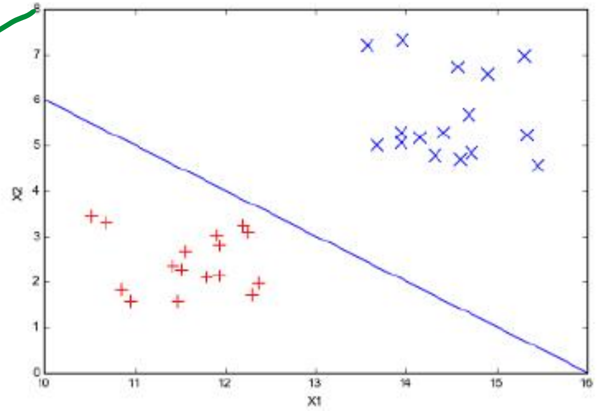
Inclusys Org Foundation
Empowering the Neurodivergent

Linearly Separable

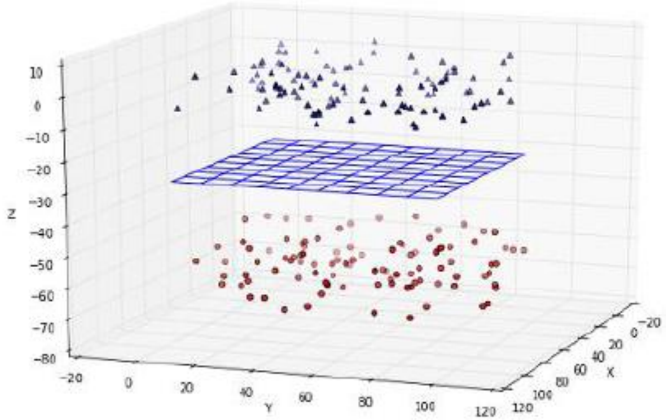
1



2



3



Hyperplanes

Hyperplane is a subspace of one dimension less than its ambient space.

- In one dimension, you can find a point separating the data
- In two dimensions, you can find a line separating the data
- In three dimensions, you can find a plane separating the data

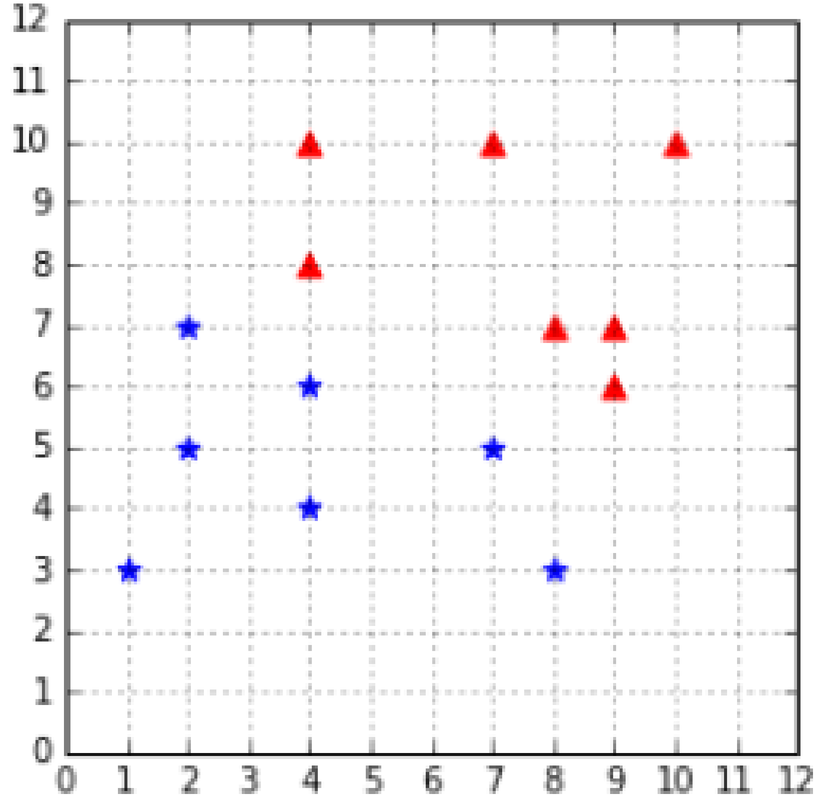
Hyperplane equation

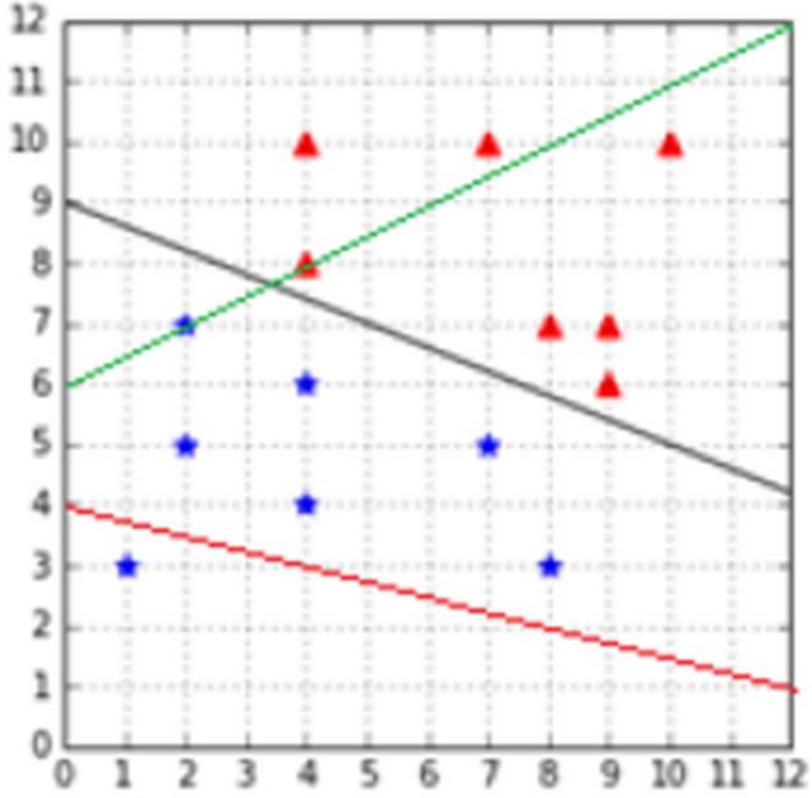
$$\boldsymbol{y = ax + b}$$

$x2 = ax1 + b \quad \Rightarrow \quad \boldsymbol{ax1 - x2 + b = 0}$

$x = (x1, x2) \quad and\ w = (a, -1) \quad \Rightarrow \quad \boldsymbol{w.x + b = 0}$

$x = (1, x1, x2) and\ w = (b, a, -1) \quad \Rightarrow \quad \boldsymbol{w.x = 0}$





Different values of **w** will give different hyperplanes

Support Vector Machines

SVM

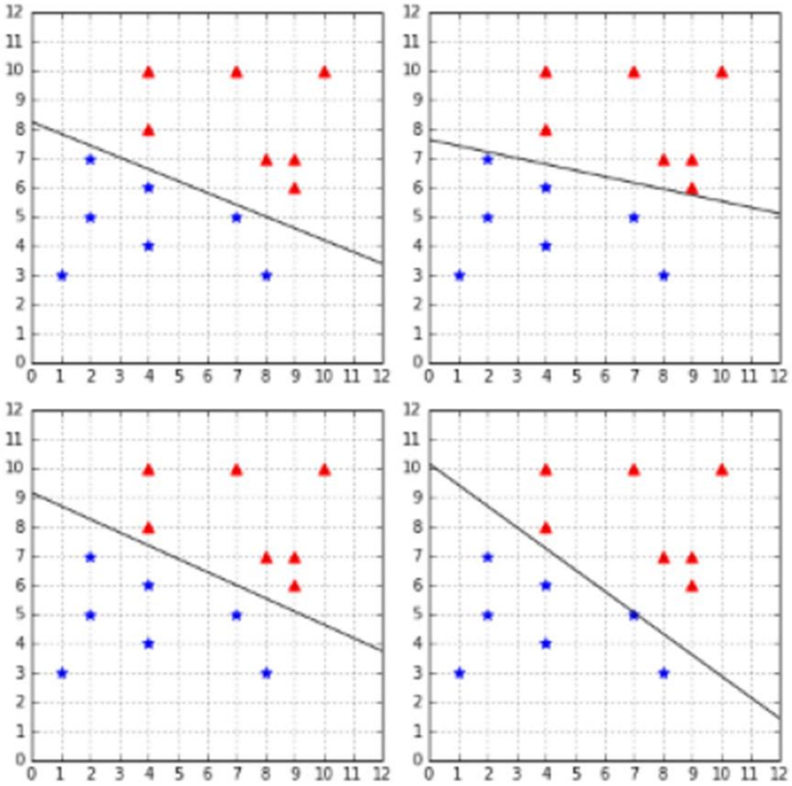
Supervised ML

Perform both Classification and Regression (mostly used for classification)

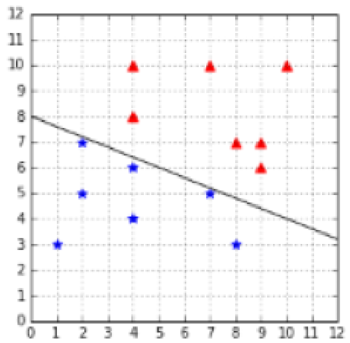
Applications:

1. Handwriting recognition
2. Face Detection
3. Text and hypertext categorization
4. Image Classification
5. Bioinformatics (protein classification and cancer classification)

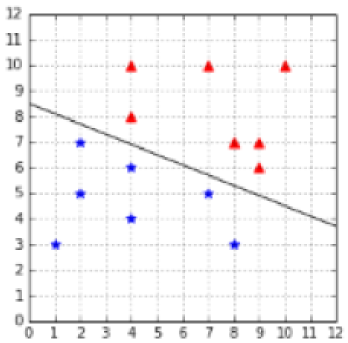
SVM tries to find the optimal hyperplane that best separates the data.



How to compare two hyperplanes?



w = (-0.4,-1)
b = 8

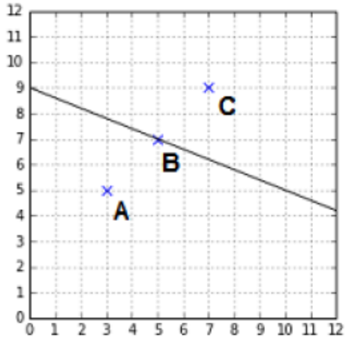


w = (-0.4,-1)
b = 8.5

Geometric margin :

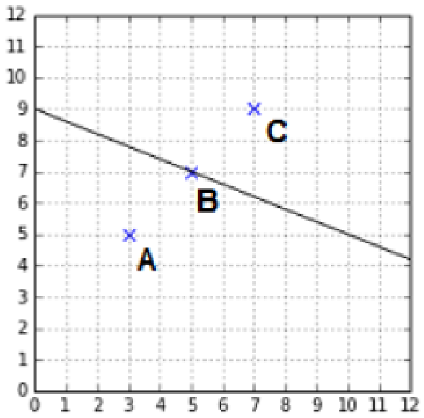
$$\gamma = y\left(\frac{\mathbf{w}}{||\mathbf{w}||} \cdot \mathbf{x} + \frac{b}{||\mathbf{w}||}\right)$$

$f = y(w.x + b)$



w = (-0.4,-1)
b = 9
Calculate w.x + b
A (1,3) ?
B (5,7) ?
C (7,9) ?

Functional Margin



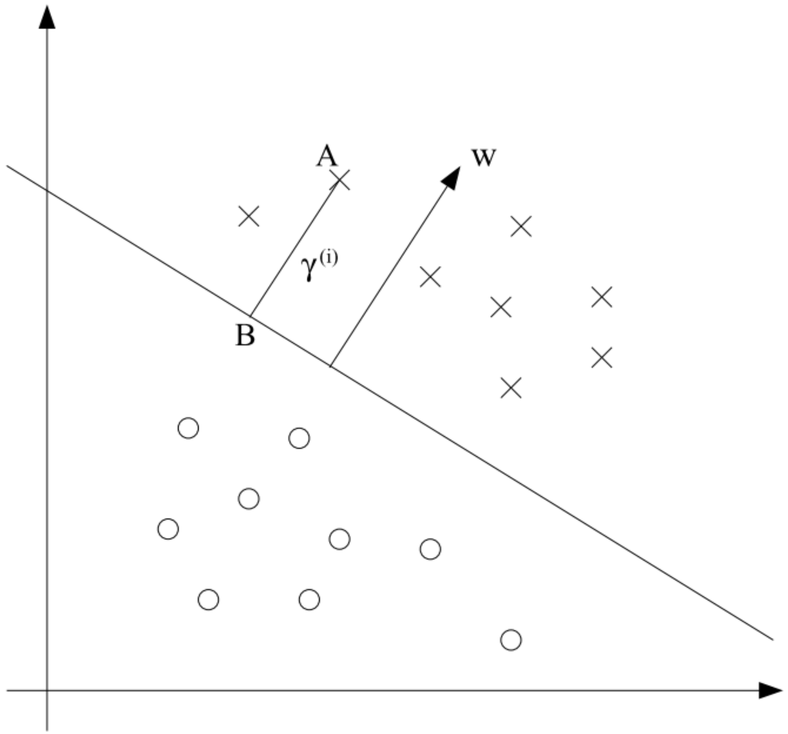
w = (-0.4,-1)
b = 9

A (1,3) **5.6**
B (5,7) **0**
C (7,9) **-2.8**

y = {-1,1}

y(w.x + b) always positive
But this is not invariant to scaling.

Geometrical margin



Geometric Margin $\boldsymbol{\gamma = y\left(\frac{w}{\|w\|} \cdot x + \frac{b}{\|\omega\|}\right)}$

Geometric margin of dataset, M = $\min\limits_{i=1,2,\ldots m} \gamma_i$

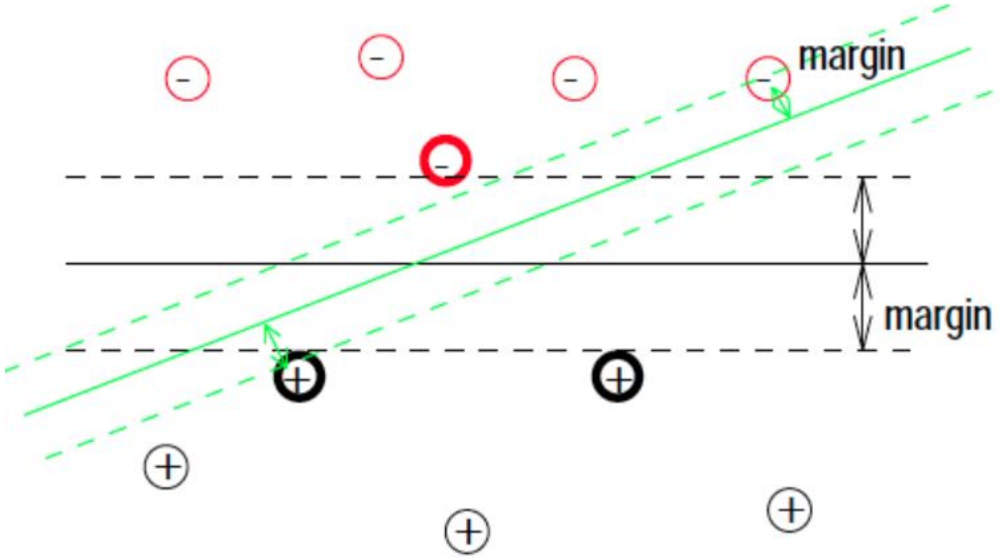
SVM optimization problem

Finding the optimal hyperplane is just a matter of finding the values of w and b for which we get the largest geometric margin.

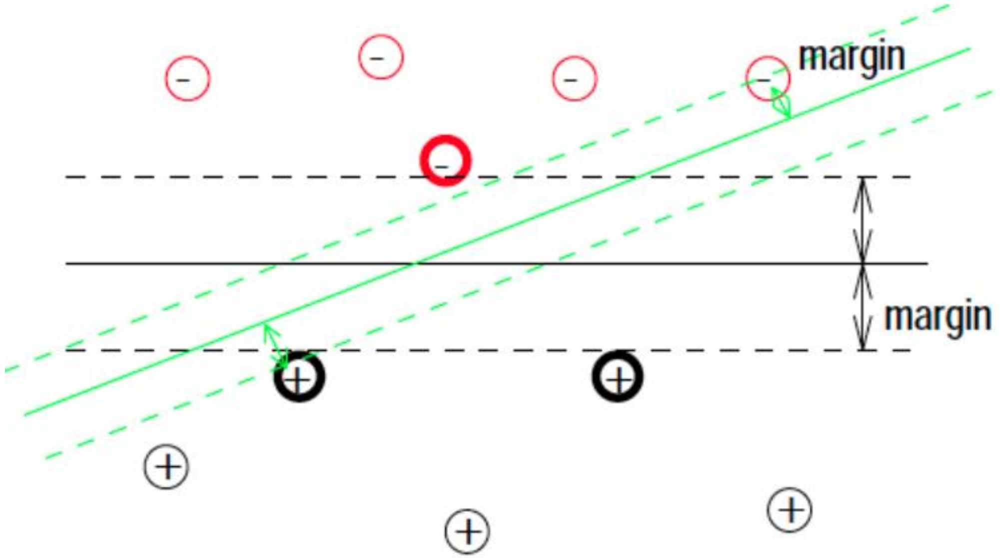
$$\underset{\mathbf{w},b}{\text{maximize}} \quad M$$
$$\text{subject to} \quad \gamma_i \geq M, \; i = 1, \ldots, m$$

$$\underset{\mathbf{w},b}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i) + b - 1 \geq 0, \quad i = 1, \ldots, m$$

Which is the optimal hyperplane?

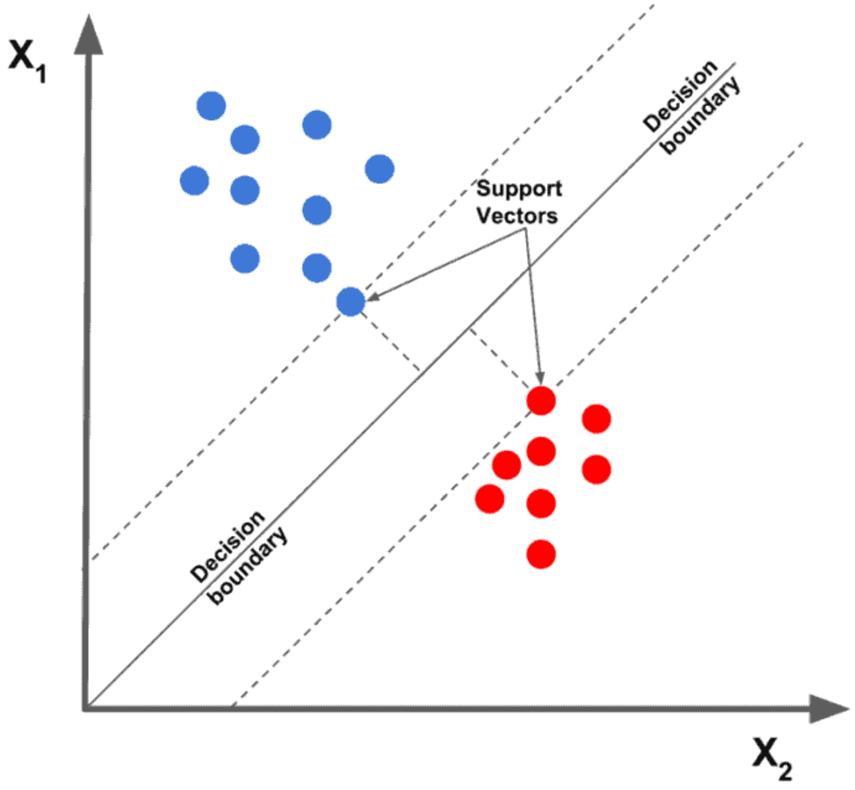


Which is the optimal hyperplane?

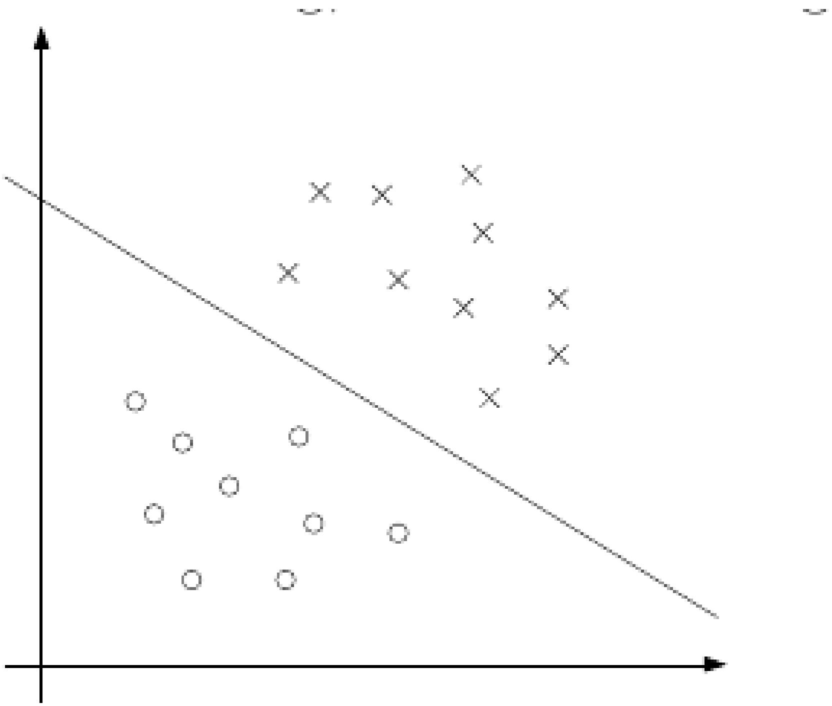


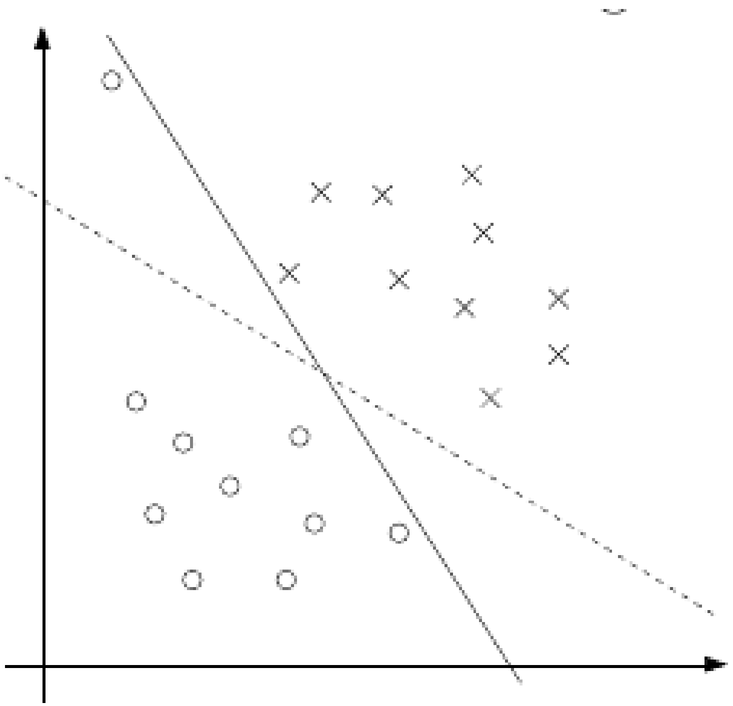
The black separating plane is better than the green one, because it has larger margins

Support Vectors

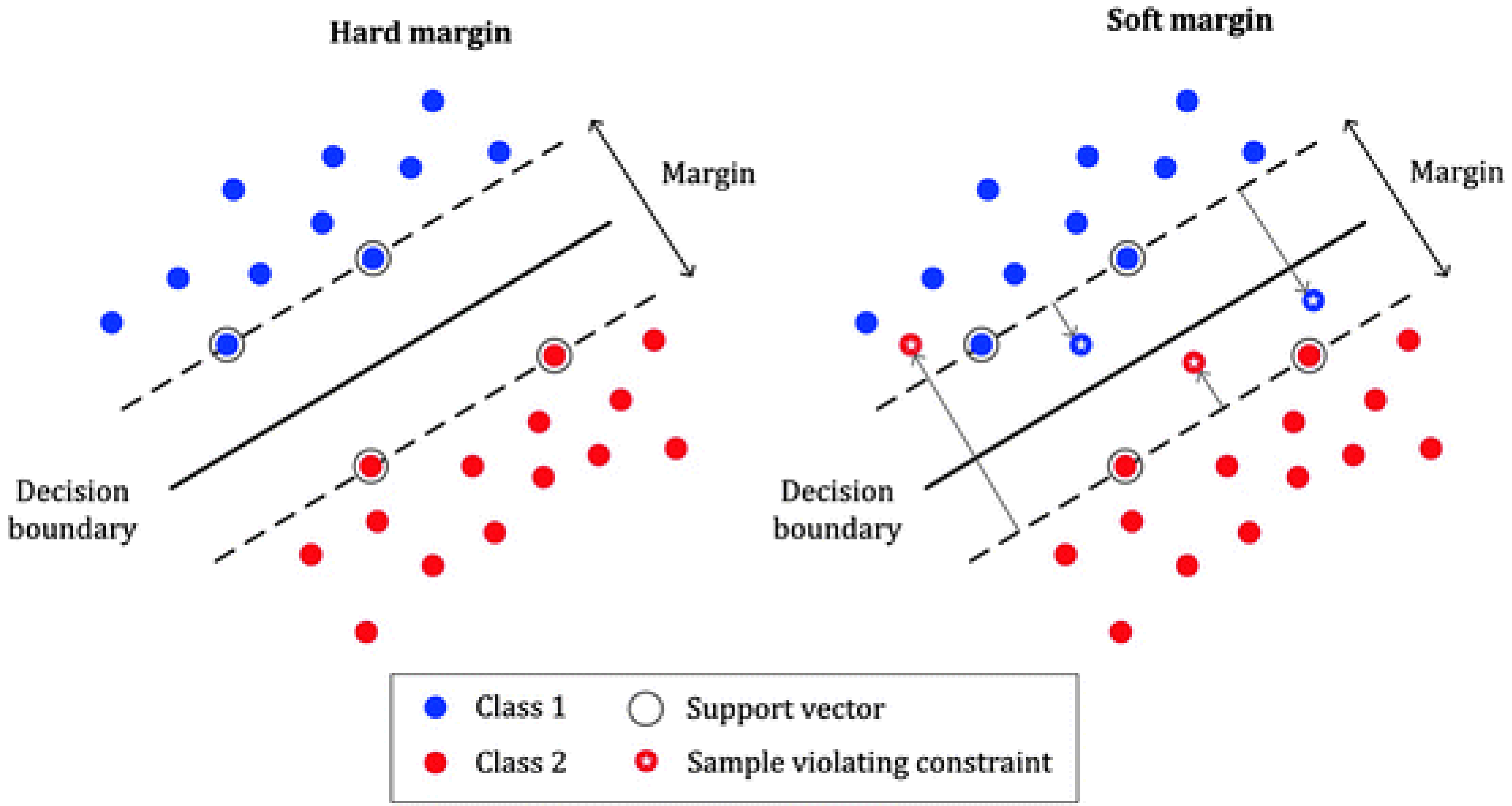


Effect of Outliers

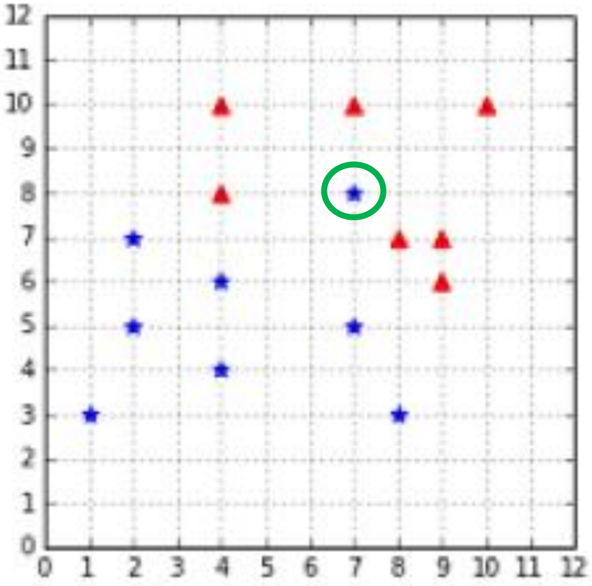




Hard and Soft Margin Classifiers

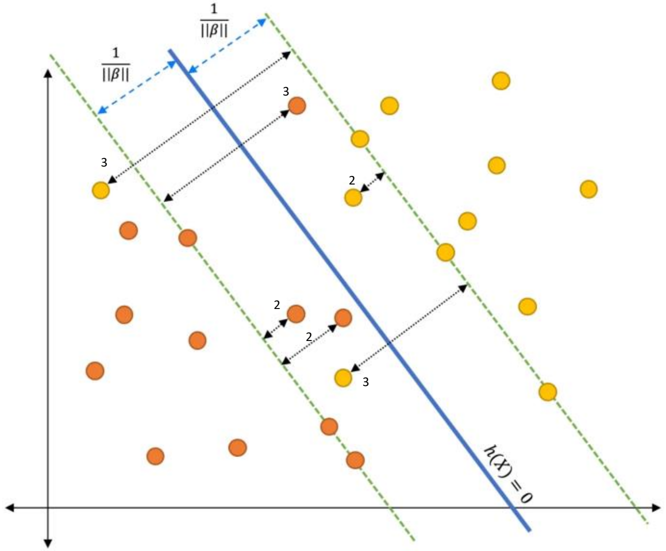


Soft Margin Formulation



$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \zeta_i$$

Soft Margin Formulation



The Slack Variable helps to define 3 types of data points:

1. if $\xi=0$ then the corresponding point ξ is on the margin or further away.
2. if $0<\xi<1$ then the point ξ is within the margin and classified correctly (Correct side of the hyperplane).
3. If $\xi\geq1$ then the point is misclassified and present at the wrong side of the hyperplane.

The ξ is the misclassification penalty. Hence we want to minimize it during optimization.

Soft Margin Formulation

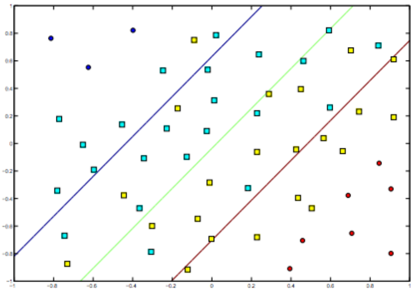
Regularization

$$\begin{aligned} \underset{\mathbf{w},b,\zeta}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\zeta_i \\ \text{subject to} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \zeta_i \\ & \zeta_i \geq 0 \quad \text{for any } i = 1,\ldots,m \end{aligned}$$

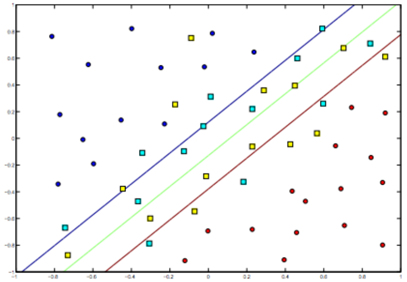
The HyperParameter C is also called as *Regularization Constant*.

- If C→0, then the loss is zero and we are trying to maximize the margin.
- If C→∞ then the margin does not have any effect and the objective function tries to just minimize the loss.

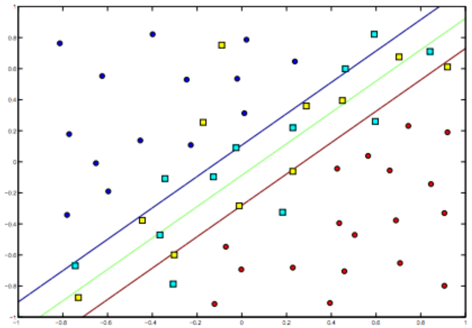
Effect of Regularization parameter



$(\lambda = 0.1)$

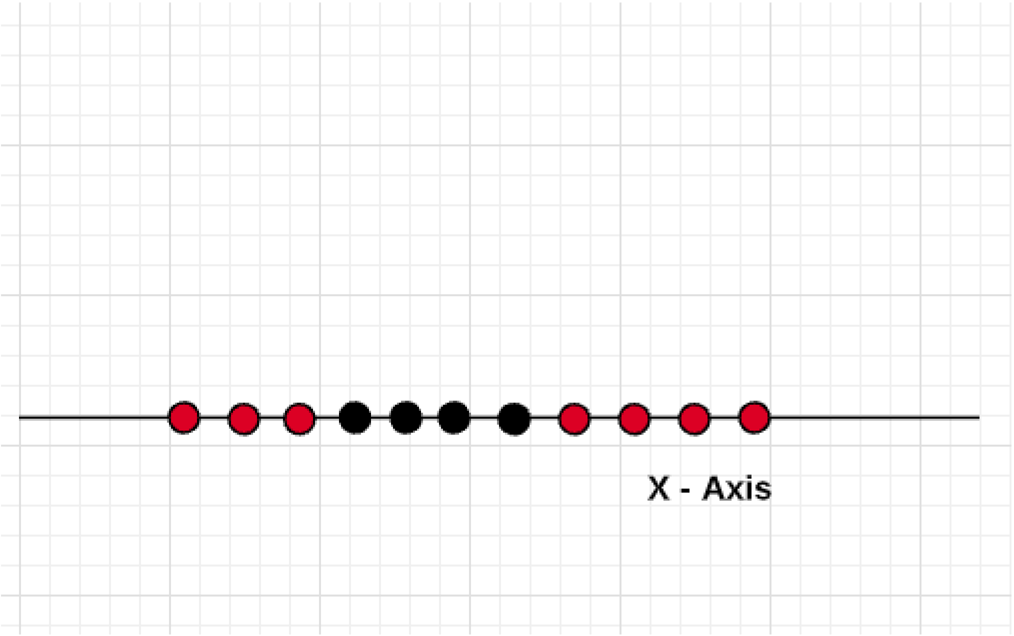


$(\lambda = 1)$

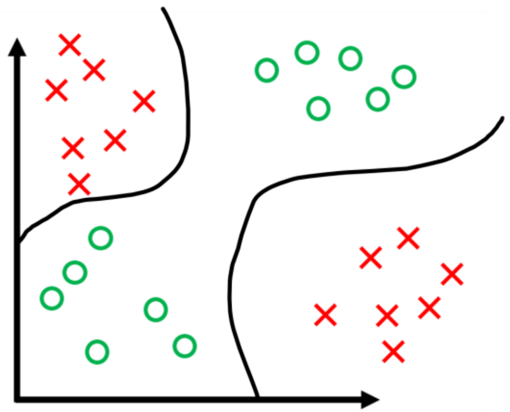


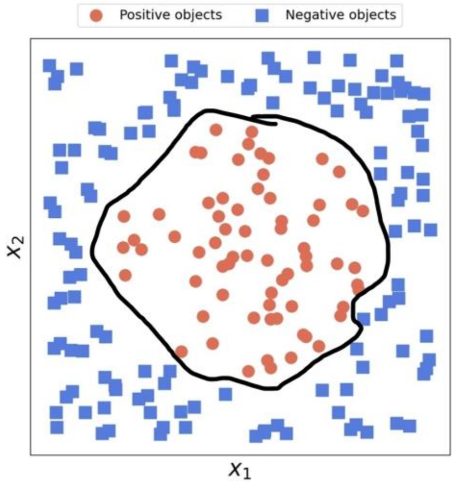
$(\lambda = 1000)$

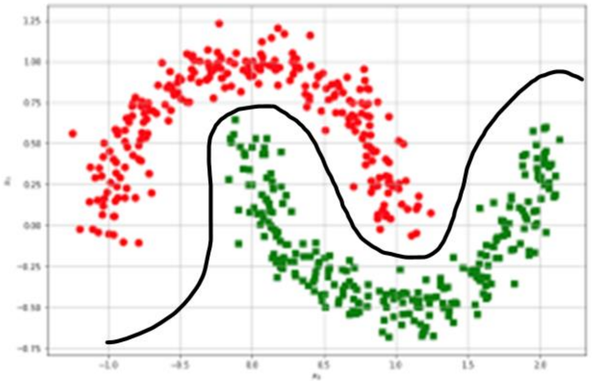
How to classify this data using SVM?



Is there a way in which a non-linear data set or data set which is linearly inseparable can be represented in another form such that the data becomes linearly separable?



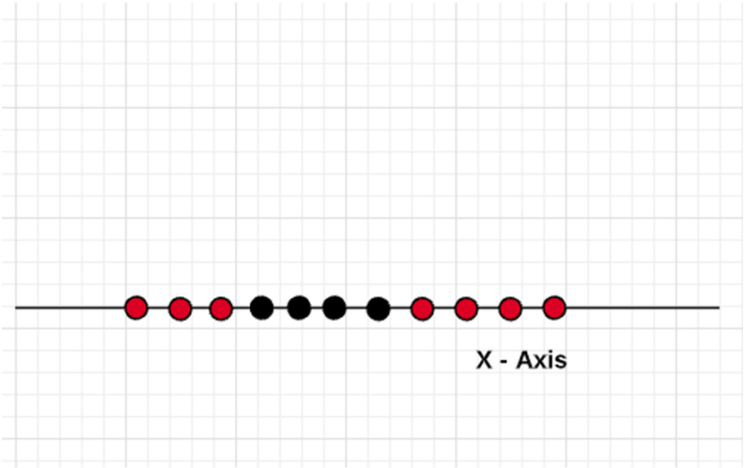




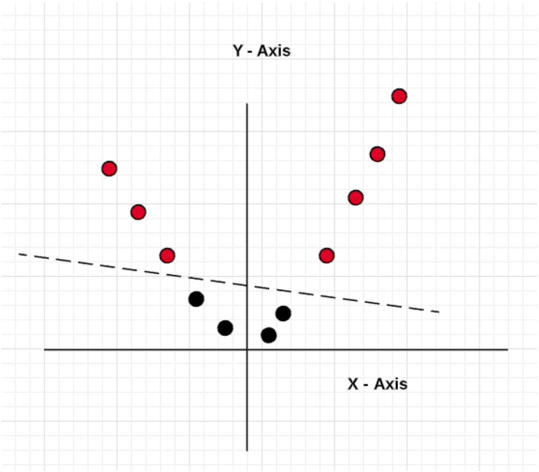
Kernel Tricks

Making Linearly Separable

The idea is to apply a function that projects / transform the data in such a manner that the data becomes linearly separable.



Transformation



Linearly inseparable data in 1D

Applying Kernel method to represent data in 2D

Kernel Methods

Kernel methods represent the techniques that are used to deal with linearly inseparable data or non-linear data set.

The idea is to create nonlinear combinations of the original features to project them onto a higher-dimensional space via a mapping function, where the data becomes linearly separable.

By using the above method, the training data can be transformed into a higher-dimensional feature space via a mapping function, and a linear SVM model can be trained to classify the data in this new feature space.

The new data can then be transformed using the mapping function and fed into the model for classification.

Kernel Trick

In SVM Formulation, we need value of dot product xi.xj only .

A kernel is a function that returns the result of a dot product performed in another space.

$$\begin{aligned} &\underset{\alpha}{\text{maximize}} && \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ &\text{subject to} && \alpha_i \geq 0, \text{ for any } i = 1, \ldots, m \\ & && \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned}$$

$$\begin{aligned} &\underset{\alpha}{\text{maximize}} && \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{subject to} && 0 \leq \alpha_i \leq C, \text{ for any } i = 1, \ldots, m \\ & && \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned}$$

An example

$x = (x1, x2)$

$\varphi(x) = \left(1, \sqrt{2}x1, \sqrt{2}x2, x1^2, x2^2, \sqrt{2}x1x2\right)$

$$\varphi(x).\varphi(z) = \left(1, \sqrt{2}x1, \sqrt{2}x2, x1^2, x2^2, \sqrt{2}x1x2\right).\left(1, \sqrt{2}z1, \sqrt{2}z2, z1^2, z2^2, \sqrt{2}z1z2\right)$$

$$= 1 + 2\mathrm{x1z1} + 2\mathrm{x2z2} + \mathrm{x1}^2\mathrm{z1}^2 + \mathrm{x2}^2\mathrm{z2}^2 + 2\mathrm{x1x2z1z2}$$

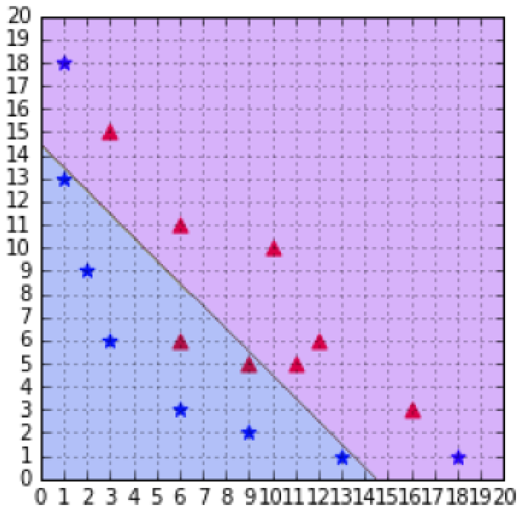
$$= (1 + \mathrm{x1z1} + \mathrm{x2z2})^2$$

$$= (1 + \mathrm{x.z})^2$$

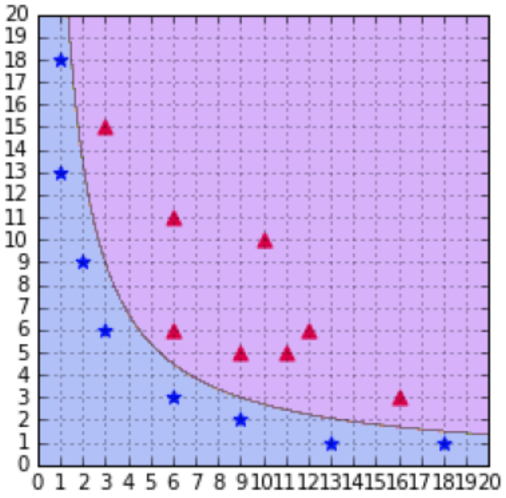
Kernel Types

1. Linear : $K(x, x') = x.x'$
2. Polynomial $K(x, x') = (x.x' + c)^d$
3. Radial Basis Function (Gaussian) $K(x, x') = \exp(-\gamma||x - x'||^2)$

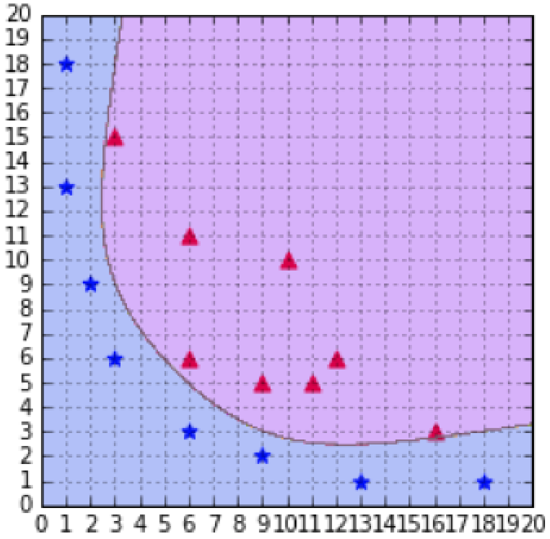
Polynomial Kernel



Degree 1

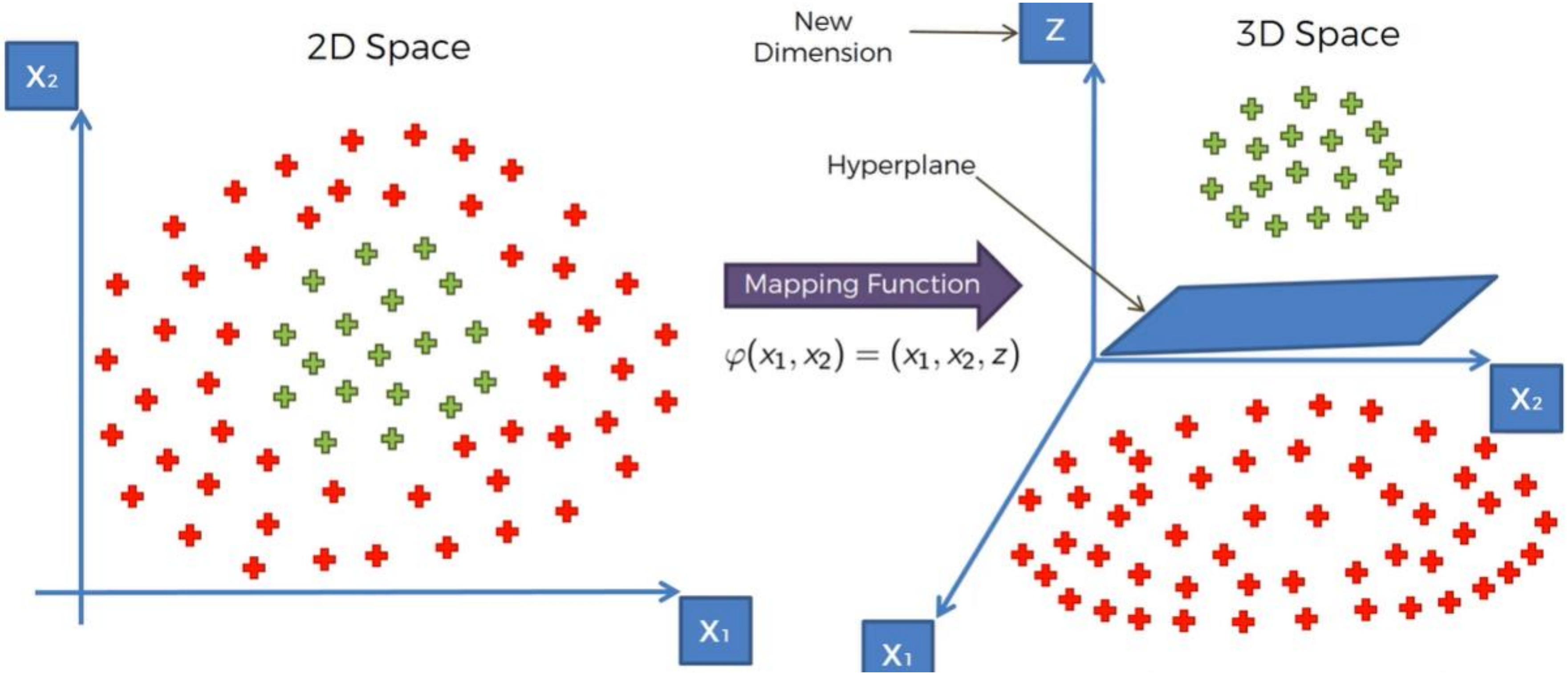


Degree 2



Degree 6

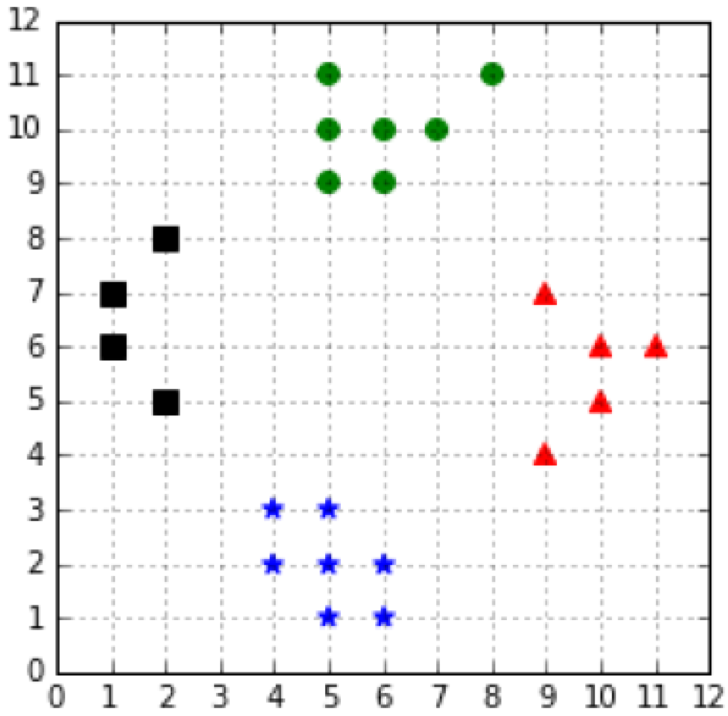
RBF



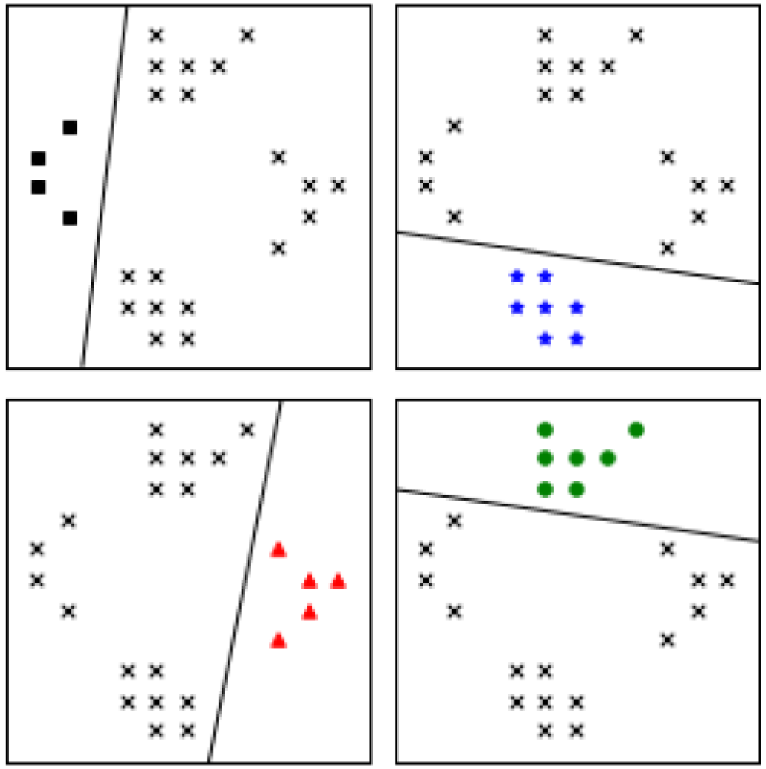
2D Space
X_2
X_1
New Dimension
Z
3D Space
Hyperplane
Mapping Function
$\varphi(x_1, x_2) = (x_1, x_2, z)$
X_2
X_1

Multiclass SVM

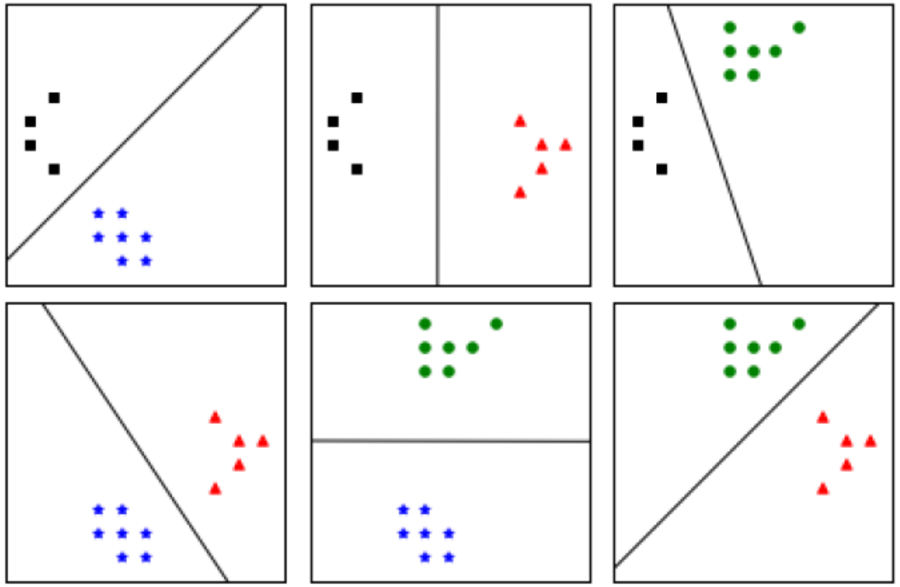
How to classify?



One against all

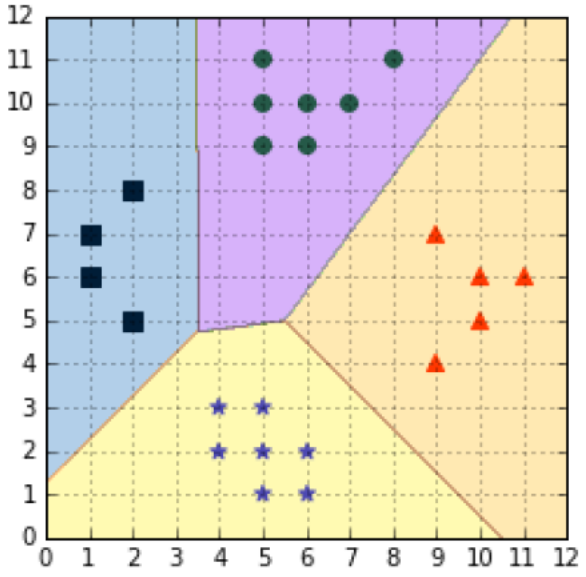


One against one



As a single optimization problem

Crammer & Singer Algorithm



Python Implementation

Implementation

……………………………………………………………

```
#Import svm model
from sklearn import svm
#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel
#Train the model using the training sets
clf.fit(X_train, y_train)
#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

Tuning Hyperparameters

- Kernel
- Gamma
- Regularization
- Type of multiclass classification

Thank You!